A Longitudinal Analysis of the ads.txt Standard

Muhammad Ahmad Bashir
Northeastern University
Boston, MA, USA
ahmad@ccs.neu.edu

Sajjad Arshad
Northeastern University
Boston, MA, USA
arshad@iseclab.org

Engin Kirda
Northeastern University
Boston, MA, USA
ek@ccs.neu.edu

William Robertson
Northeastern University
Boston, MA, USA
wkr@ccs.neu.edu

Christo Wilson
Northeastern University
Boston, MA, USA
cbw@ccs.neu.edu

ABSTRACT

Programmatic advertising provides digital ad buyers with the convenience of purchasing ad impressions through Real Time Bidding (RTB) auctions. However, programmatic advertising has also given rise to a novel form of ad fraud known as domain spoofing, in which attackers sell counterfeit impressions that claim to be from high-value publishers. To mitigate domain spoofing, the Interactive Advertising Bureau (IAB) Tech Lab introduced the ads.txt standard in May 2017 to help ad buyers verify authorized digital ad sellers, as well as to promote overall transparency in programmatic advertising.

In this work, we present a 15-month longitudinal, observational study of the ads.txt standard. We do this to understand (1) if it is helping ad buyers to combat domain spoofing and (2) whether the transparency offered by the standard can provide useful data to researchers and privacy advocates.

With respect to halting domain spoofing, we observe that over 60% of Alexa Top-100K publishers that run RTB ads have adopted ads.txt, and that ad exchanges and advertisers appear to be honoring the standard. With respect to transparency, the widespread adoption of ads.txt allows us to explicitly identify over 1,000 domains belonging to ad exchanges, without having to rely on crowdsourcing or heuristic methods.

However, we also find that ads.txt is still a long way from reaching its full potential. Many publishers have yet to adopt the standard, and we observe major ad exchanges purchasing unauthorized impressions that violate the standard. This opens the door to domain spoofing attacks. Further, ads.txt data often include errors that must be cleaned and mitigated before the data is practically useful.

ACM Reference Format:
Muhammad Ahmad Bashir, Sajjad Arshad, Engin Kirda, William Robertson, and Christo Wilson. 2019. A Longitudinal Analysis of the ads.txt Standard. In *Internet Measurement Conference (IMC '19), October 21–23, 2019, Amsterdam, Netherlands.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3355369.3355603

IMC '19, October 21–23, 2019, Amsterdam, Netherlands

ACM ISBN 978-1-4503-6948-0/19/10…$15.00
https://doi.org/10.1145/3355369.3355603

1 INTRODUCTION

Despite being the primary source of funding for free content online, the online display advertising ecosystem is a $127 billion enigma [78]. Researchers and industry groups have documented hundreds of different companies taking part in the ecosystem with a plethora of different business models, ranging from trackers, to data brokers, to market makers, to advertisers [11, 15, 16, 34, 60]. The advent of *programmatic* advertising based on Real Time Bidding (RTB) auctions has only increased the complexity of the ecosystem, by enabling more players to participate in the marketplace, while also accelerating the movement of data and *impressions* to milliseconds speeds.

The complexity, scale, and opacity of the ad ecosystem create opportunities for various kinds of fraud. While *click* and *impression fraud* are longstanding problems [24, 27, 87, 88], RTB in particular has opened the door to a novel fraud known as *domain spoofing* [18, 50, 51]. In this attack, the fraudster creates fake bid requests for impressions that were purportedly generated by visitors to high-value *publishers* (e.g., CNN or YouTube). Advertisers bid highly to show their ads on these valuable publishers, but the ads end up appearing on low-value websites, or nowhere at all, while the fraudster collects the profit. Attackers can earn millions of dollars per day spoofing bid requests [18, 64].

The fundamental issue that enables domain spoofing is the opacity of the RTB ecosystem: advertisers cannot tell which auctioneers are *authorized* to sell impression inventory from a given publisher. This lack of transparency gives attackers the ability to spoof inventory from any publisher. To address this problem, the Interactive Advertising Bureau (IAB) Tech Lab introduced the ads.txt standard [80] in May 2017. ads.txt is designed to rectify this transparency problem by allowing publishers to state, in a machine-readable format, which auctioneers are authorized to sell their impression inventory [41]. To opt-in to the standard, a publisher must place a file named /ads.txt at the root of their website; auctioneers and advertisers can then download the file and verify the authenticity of bid requests.

In addition to helping mitigate domain spoofing, the ads.txt standard is potentially of interest to researchers and privacy advocates. The opacity of the online advertising ecosystem has long frustrated attempts to understand which third-parties are part of the ecosystem, as well as the role of each third-party (e.g., tracker, advertiser, auctioneer, etc.) [12, 15]. The practical consequence of

this opacity is that users have grown suspicious of online advertisers and their privacy practices [5, 61, 92]. ads.txt fundamentally changes the landscape, by making it explicit which third-party domains in a given first-party context are *ad exchanges* (i.e., auctioneers). In aggregate, ads.txt data has the potential to reveal, for the first time, the relationships between publishers, ad exchanges, and advertisers.

In this study, we take the first step towards measuring and quantifying the landscape revealed by ads.txt-compliant publishers. Our study aims to answer two basic questions:

(1) *Are members of the online ad ecosystem complying with the* ads.txt *standard?* This includes adoption of the standard by publishers, as well as enforcement (or lack thereof) of the standard by ad exchanges and advertisers when bidding on impressions.
(2) *How useful is* ads.txt *as a transparency mechanism?* This includes the scope, specificity, and correctness of the data contained in ads.txt files.

To answer these questions, we crawled ads.txt files from Alexa Top-100K websites on a monthly basis between January 2018 and April 2019. We focus on these websites because their impressions are valuable, and thus they have the strongest incentive to adopt ads.txt. We also conducted monthly crawls of the Alexa Top-100K websites to gather information about the ad exchanges and advertisers that each website interacted with. This data allows us to observe whether auctioneers and advertisers appear to be in compliance with the rules stipulated in publishers' ads.txt files.

Although we study compliance with the ads.txt standard to see its *potential* for combatting fraud, we are not able to measure the effect of the standard on limiting *actual* fraud. Quantifying the direct impact of ads.txt on domain spoofing fraud is challenging, and would necessitate either (1) becoming a publisher and conducting active experiments, or (2) partnering with a major ad exchange to measure their internal datasets.

Through this study, we make the following key contributions and findings:

- We present the first large-scale, longitudinal study of the ads.txt standard. We observe that as of April 2019, 20% of Alexa Top-100K websites have adopted the standard, which rises to 62% when we only consider websites that display ads via RTB auctions. This demonstrates that ads.txt has achieved impressive adoption since it was introduced in May 2017.
- With respect to compliance, we find that the vast majority of RTB ads in our sample were bought from authorized sellers. This suggests that ad exchanges and advertisers are complying with the standard. However, we also see that domain spoofing is still possible, because major ad exchanges still accept impression inventory from publishers that have not adopted ads.txt. Further, we document cases where major ad exchanges purchased impressions from unauthorized sellers, in violation of the standard.
- With respect to transparency, ads.txt allows us to identify the third-party ad exchanges on ~62K publishers that run RTB ads and isolate 1,035 unique domains belonging to ad exchanges. That said, we also find that ads.txt data has a variety of imperfections, and we develop methods to mitigate these deficiencies.

Open Source. As a service to the community, we have open-sourced the data from this project. This includes 26 snapshots of the ads.txt files from Alexa Top-100K publishers between January 2018 and April 2019, cleaned list of authorized sellers, associated inclusion chains, and a list of ad exchange domains clustered by their respective parent organizations. The data is available at:
https://personalization.ccs.neu.edu/Projects/Adstxt/

Organization. Our study is organized as follows. In § 2, we define key terms and explain the ads.txt standard. In § 3 we explain how we crawled and cleaned the data used throughout this study. In § 4 we analyze ads.txt adoption from the perspective of publishers and ad exchanges, while in § 5 we investigate compliance with the standard. We briefly survey related work in § 6, discuss limitations in § 7, and conclude our findings in § 8.

2 BACKGROUND

We begin by briefly introducing the programmatic online advertising ecosystem, defining key terms, discussing the rationale behind ads.txt, and discussing the ads.txt standard in detail.

2.1 RTB Overview

Over time, the mechanisms for selling and buying *impressions* have become *programmatic* via *Real Time Bidding (RTB) auctions*. In industry parlance, *publishers* (i.e., websites/apps that distribute media to consumers) aim to monetize their *impression inventory* (i.e., the attention of people visiting their service) by selling it to advertisers. At a high-level, whenever a person visits a publisher, their browser will contact an *ad exchange* that serves as the auctioneer. The ad exchange solicits bids on the impression from advertisers, who have just milliseconds to respond. The ad exchange then redirects the user's browser to the winning advertiser so they may serve an ad. Programmatic advertising is estimated to account for 83% of all US digital display advertising as of 2020 [33]. It is popular because it increases fluidity in the advertising market, as well as allowing publishers to increase their revenue (in theory) by selling their inventory to the highest bidders on-demand.

Although RTB auctions are conceptually simple, they are complex in practice. With respect to the *sell-side*, publishers form business relationships with ad exchanges and other *Supply-Side Platforms (SSPs)* that facilitate the selling of impressions. Examples of ad exchanges include the Google Marketing Platform (formerly Doubleclick), Rubicon, and OpenX. With respect to the *buy-side*, *Demand-Side Platforms (DSPs)* represent advertisers by purchasing impressions to implement their campaigns. Examples of DSPs include Criteo, Quantcast, and MediaMath. Note that many companies offer both seller- and buyer-side products (e.g., Google and Rubicon), complicating their role in the ecosystem. Furthermore, impressions can be resold after they are won, i.e., the winner of an RTB auction may be another ad exchange, which will then hold another auction, etc. This can lead to long *chains* of transactions that separate the true source of an impression from the DSP that eventually serves an ad.

2.2 Ad Fraud and Spoofing

The online ad ecosystem has long been plagued with fraud, generating estimated losses of $8.2 billion per year in 2015 [49]. The most well-known forms are *impression fraud* and *click fraud* [27, 73, 87]. In this scheme, the attacker creates a seemingly-legitimate publisher and contracts with ad exchanges to sell their impressions. The attacker then earns revenue by directing fraudulent traffic to their own publisher. We discuss prior work on these forms of fraud in § 6.

The rise of programmatic advertising has created an opportunity for a different type of fraud known as *domain spoofing* or sometimes *inventory counterfeiting* [18, 50, 51]. In this scheme, the attacker generates bid requests that are supposedly for impressions on a high-value publisher (e.g., CNN or The New York Times), when in reality these impressions are either (1) entirely fabricated or (2) actually generated from a low-value publisher (which is often controlled by, or collaborates with, the attacker). Attackers can implement spoofing attacks by creating or compromising an SSP, or (in some cases) simply by setting up an illegitimate publisher. The attacker can make their spoofed inventory harder to detect by mixing it with legitimate inventory [88].

2.3 A Brief Intro to ads.txt

The fundamental flaw in the programmatic advertising ecosystem that enabled domain spoofing is that legitimate ad exchanges and DSPs had no way of knowing which ad exchanges/SSPs were *authorized* to sell impression inventory from a given publisher. This lack of transparency gave attackers the ability to spoof inventory from any publisher.

To combat spoofing, the Interactive Advertising Bureau (IAB) Tech Lab, which is a non-profit trade association for online advertisers, introduced the ads.txt standard [80]. The standard is designed as a first step towards rectifying the transparency issues that allowed spoofing to flourish, by allowing publishers to state, in a machine-readable format, which SSPs and ad exchanges are authorized to sell their impression inventory. To be compliant with the standard, ad exchanges and SSPs are supposed to reject inventory they are not authorized to sell, while DSPs are not supposed to buy inventory from unauthorized sellers.

ads.txt 1.0 was introduced in May 2017 [80], and the latest 1.0.2 standard was published in March 2019 [41]. Google announced that by December 2018, DSPs in their exchange would purchase impressions that were authenticated via ads.txt by default [39, 46], i.e., a DSP would need to opt-out of the security measure if they wanted to purchase unauthenticated impressions. Google runs one of the largest ad exchanges [15], which created a strong incentive for publishers to adopt ads.txt by the end of 2018 if they wanted their inventory to be purchasable by all DSPs in the auction.

ads.txt is just the first step towards combatting domain spoofing fraud, and is by no means perfect [31]. The IAB is working on improving the ads.txt standard in conjunction with the OpenRTB 3.0 specification [56] by providing an upgrade called ads.cert [28]. Through ads.cert, publishers will be able to cryptographically sign bid requests to authenticate their inventory.

```
# CNN.com/ads.txt
google.com, pub-7439281311086140, DIRECT, f08c47fec0942fa0
rubiconproject.com, 11078, DIRECT, 0bfd66d529a55807
c.amazon-adsystem.com, 3159, DIRECT # banner, video
openx.com, 537153334, DIRECT # banner
openx.com, 540038342, DIRECT, a698e2ec38604c6 # banner
pubmatic.com, 156565, RESELLER, 5d62403b186f2ace # banner
pubmatic.com, 156599, DIRECT, 5d62403b186f2ace # banner
```

Listing 1: Example ads.txt taken from cnn.com on May 11, 2019 (and edited for brevity).

2.4 ads.txt File Format

Much like the robots.txt exclusion standard [52], the ads.txt standard is instantiated by including a text file named /ads.txt at the root of a website. Listing 1 shows an example ads.txt file for illustrative purposes. ads.txt files obey a simple, line-oriented format; in keeping with the IAB specification [41], we refer to each line as a *record*. Each record contains three or four comma-separated fields that authorize a given SSP/ad exchange to sell impression inventory on behalf of the given publisher. The fields are:

(1) **Seller Domain**: A domain name specifying the SSP or ad exchange that the publisher is authorizing to sell their impression inventory.
(2) **Publisher ID**: A string that uniquely identifies the publisher's account within the ad system hosted by the company in field 1.
(3) **Relationship**: Either "DIRECT" or "RESELLER" depending on whether the publisher is the contractual owner of the advertising account in field 2 (former) or that the publisher has contracted with a third-party to manage the account (latter).
(4) **Certification Authority ID** (Optional): An ID that uniquely corresponds to the company in field 1. As of this writing, these IDs are assigned by the Trustworthy Accountability Group.[1]

Every <seller, publisher ID, relationship> triple uniquely defines a business relationship between the given seller and the publisher who authored the ads.txt file. Note that a given seller/publisher pair may have multiple business relationships, each encoded as a different record in the ads.txt file. As shown in Listing 1, this may happen if the publisher has multiple accounts with the seller (field #2 varies) and/or because the publisher has DIRECT and RESELLER relationships with the seller (field #3 varies).

ads.txt files may also contain comments, delimited by the "#" character. These may appear on their own line or at the end of record lines. Further, ads.txt files may contain additional meta-data that appears in a "variable=value" format.[2] In our dataset (described in § 3), we observe that this meta-data is rare, and we ignore it in this study.

The most confusing aspect of the ads.txt standard is that the seller domains listed in field #1 are not necessarily the domains that host ad auctions. For example, Google specifies that its seller domain is google.com, even though the actual auctions are hosted

[1] https://www.tagtoday.net/
[2] Through the "variable=value" record, author of the ads.txt file can provide their contact information or point towards a subdomain that operates its own ads.txt file.

```
# Incorrect format, less than 3 comma separated fields
google.com – pub-7439281311086140, DIRECT
# Invalid seller domain, misspelled rubiconproject.com
rubicnproject.com, 17380, DIRECT, 0bfd66d529a55807
# doubleclick.net is incorrect, should be google.com
doubleclick.net, pub-7439281311086140, DIRECT
```

Listing 2: Example `ads.txt` containing different classes of errors in each record.

at `doubleclick.net`. Each SSP/ad exchange defines what domain should be placed in field #1 to authorize them.

3 METHODOLOGY

The primary goal of our study is to examine the `ads.txt` standard. In particular, we want to monitor publishers' adoption of the standard, the involvement of authorized sellers (exchanges/SSPs), and compliance with the standard by buyers (DSPs). In this section, we outline how we collected and cleaned `ads.txt` data. Then we describe how we collected inclusion trees (inclusion of resources) from websites to determine compliance with the `ads.txt` standard.

3.1 Collection of `ads.txt` Data

The most crucial dataset for our study is `ads.txt` files from publishers. To obtain this data, we started crawling the Alexa Top-100K websites on January 15, 2018. Up until December 1, 2018, we repeated our `ads.txt` crawl every 15 days. After that, we crawled once every 30 days (on the 1^{st} of each month). The latest snapshot used in this study is from April 1, 2019. Overall, we performed 26 crawls.

Subsequent to the start of our data collection, Scheitle et al. [83] and others [77, 82] published compelling analyses that document instabilities in the Alexa ranking. Considering these results, from October 15, 2018 onwards, we started updating the list of target websites in our crawl: before each crawl, we fetched the latest Alexa Top-100K list, computed the union of it and our existing list of target websites, and crawled the result. Subsequently, our sample size grew from 100K websites on January 15, 2018 to 240K on April 1, 2019.

According to the IAB standard, the `ads.txt` file must be placed at the root of a given domain. We used Python's `requests` module to fetch the `ads.txt` files: for each publisher p from the Alexa Top-100K, we accessed the `/ads.txt` URL from p's root. We sent a valid `User-Agent` with each request. We were able to crawl all the target websites within 2–3 hours by parallelizing across a 16-node cluster at Northeastern University.[3]

3.1.1 Parsing and Cleaning. To facilitate analysis, we parsed all of the `ads.txt` files gathered by our crawler. In theory, `ads.txt` files are supposed to obey the IAB specified format outlined in § 2.4; in practice, we observed many files with errors, which necessitated that we develop a custom approach for parsing and validating `ads.txt` files.

We observed that publishers made a variety of mistakes in their `ads.txt` files, of which we highlight three examples in Listing 2. Some records, such as the first in Listing 2, contain syntactic errors,

[3]The IAB provides a prototype crawler to fetch `ads.txt` files [48]. We use ideas from there to build our own custom crawler for large-scale crawling and post-processing.

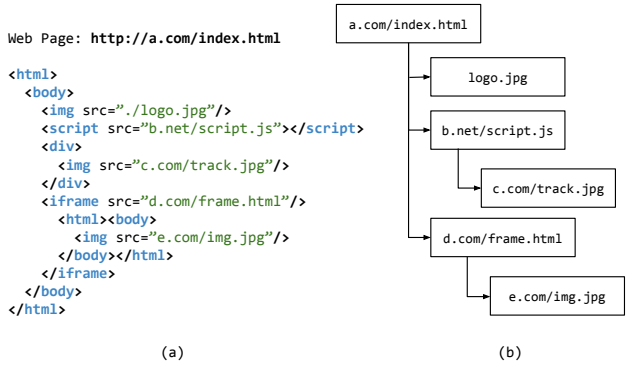

Figure 1: Example DOM tree with corresponding *inclusion tree*.

i.e., they do not obey the formatting specification. Other records contained semantic errors. For example, the second record in Listing 2 is in the correct format, but the seller is incorrect: it is supposed to be `rubiconproject.com`, but is `rubicnproject.com` instead. The third record in Listing 2 illustrates an even subtler error, where the seller domain has been accidentally replaced by a related, but incorrect, domain. In this case, the seller should be `google.com`, but was mistakenly added as `doubleclick.net`.

We used a multi-stage filtering process to remove records with syntax errors and some semantic errors. First, we discarded all records that did not conform to the `ads.txt` specification (e.g., the first record in Listing 2). Second, we extracted all 2,381 unique seller domains S from the syntactically valid records in our dataset. Third, to identify semantically invalid domains (like the second record in Listing 2), we queried each domain in the WHOIS database. We were able to find WHOIS data for 1,035 of the seller domains. To make sure that we did not have any false negatives (i.e., the WHOIS crawl failed to fetch data for a valid seller domain), we also performed DNS resolution on all the negative samples. None of the domains in the negative sample had a successful resolution. Therefore, unless mentioned otherwise, we only consider the 1,035 seller domains S_v in our analysis. Further, we disregard all records containing the 1,346 unresolvable seller domains.

Our filtering method cannot identify semantic errors like in the third record in Listing 2 because, in these cases, the erroneous domains are valid and resolvable. As we discuss in § 4.2, we estimate that ~20% of the unique sellers in our dataset are the result of such errors, but these low-frequency sellers end up having very limited impact on our analysis.

3.2 Inclusion Trees

To assess compliance with the `ads.txt` standard on an `ads.txt`-enabled publisher, we need to examine which sellers and buyers were involved in serving ads through RTB auctions. To accomplish this we rely on *inclusion trees*, which are a data structure introduced by Arshad et al. [8] that have subsequently been used for several studies of web security [57] and online advertising [12, 13, 15]. Inclusion trees capture the semantic relationships between resource

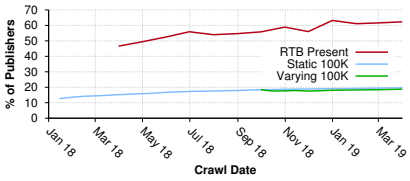

Figure 2: ads.txt adoption by Alexa Top-100K publishers over time.

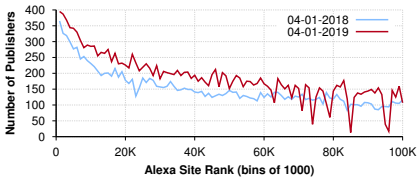

Figure 3: Publisher adoption over alexa ranks.

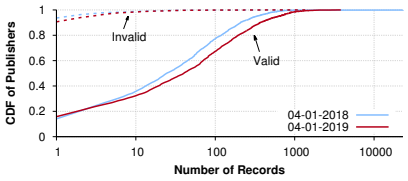

Figure 4: Number of ads.txt records per publisher.

inclusions in a website. Figure 1 shows an example Document Object Model (DOM) tree and its corresponding inclusion tree.

We cannot rely on the DOM to determine how an ad was shown because it encodes syntactic structures, rather than the semantic relationships between resource inclusions. For example, as shown in Figure 1, the resources from b.net and c.com have no obvious relationship encoded in the DOM, but the inclusion tree correctly marks that c.com's resource was included by b.net's script.

Furthermore, analyzing HTTP request headers to determine resource inclusions is also insufficient. Specifically, the Referer field may be inaccurate when JavaScript from a third-party is included in a first-party context. Bashir et al. demonstrated that up to 48% of resource inclusions in a typical, crawled dataset can have inaccurate Referer (i.e., the resource was requested by third-party JavaScript, but the Referer was assigned to the first-party) [15].

We were able to capture inclusion trees for a website using the Chrome Debugging Protocol [19]. This protocol grants us fine-grained access to Chrome's internals without the need to instrument the browser's source code. To capture dynamic inclusions, we used scriptParsed events in the Debugger domain, and requestWillBeSent and responseReceived events in the Network domain. Through scriptParsed, we can track JavaScript triggered by remote and inline scripts, whereas requestWillBeSent and responseReceived are used to observe any further resource requests. We capture iframe inclusions by collecting frameNavigated events in the Page domain.

3.2.1 Collecting Resource Inclusions. Using the technique from § 3.2, we repeatedly drove a Chrome browser to collect resource inclusions for all the publishers from the ads.txt crawl. These crawls were done right after each ads.txt crawl finished (see § 3.1). In particular, for each publisher p in the dataset, the crawler visited the homepage for p, then iteratively crawled 15 randomly selected links that pointed to p. During these crawls, we presented a valid User-Agent, scrolled pages to the bottom, and waited ~10 seconds between subsequent page visits.

Once we have collected inclusion trees from publishers, we decompose them into *inclusion chains* to facilitate analysis. For a given inclusion tree (corresponding to a single visit of a webpage), the chains are simply all of the root-to-leaf paths in the tree.

Crawling Tool. The tool we used to crawl inclusion chains in this study is publicly available at:

`https://github.com/sajjadium/DeepCrawling`

3.2.2 Detecting Ads. The last step in our methodology is identifying all of the inclusion chains that correspond to the serving of an ad. We do this by applying a series of filters: first, we eliminate all chains where the final resource is not an image. Second, we filter out chains where the final image is $\leq$ 50×50 pixels.[4] Finally, we filter out chains that include zero requests to a URL that matches a rule in EasyList [32]. This last step allows us to separate benign images from advertisements by ensuring that a known advertising-related URL was involved in serving the image.

4 ADOPTION OF ADS.TXT

In this section, we analyze the adoption of the ads.txt standard over our 15-month study. We examine adoption trends from the perspective of Alexa Top-100K publishers and top sellers that appear in the ads.txt files.

4.1 Publisher's Perspective

We begin by examining the ads.txt standard from the perspective of publishers, starting with the adoption of the standard by Alexa Top-100K websites over time. The *Static 100K* line in Figure 2 shows adoption by a static set of Alexa Top-100K websites that was sampled in January 2018. The Varying 100K line shows adoption by a dynamic set of Alexa Top-100K websites that grows over time to incorporate newly popular sites (see § 3.1). In January 2018, we observed 12.7% of websites adopting the standard, which grew steadily to 19.7% in April 2019. Adding new, popular websites over time had negligible impact on our results. Further, our observations match those of Lukasz Olejnik, an independent researcher who has also been tracking ads.txt adoption [69].

Although adoption of ads.txt by Alexa Top-100K websites is modest overall, this baseline is too liberal since it includes websites that (1) do not display ads or (2) do not display ads via ad exchanges (e.g., Facebook, YouTube). There is no reason for these classes of websites to adopt ads.txt. To account for this, we isolate the set of websites W_{RTB}, that appear to be displaying ads via RTB auctions, from our complete set of crawled websites W. At a high-level, website $w \in W$ is also a member of W_{RTB} if we observe $\geq$1 inclusion chain rooted at w that includes $\geq$1 requests to a known ad exchange. We derive this list of known ad exchanges from the ads.txt data itself; see § 5.2 for further details.

The *RTB Present* line in Figure 2 shows adoption of ads.txt over time by websites in W_{RTB}. We observe that adoption has increased from 46.6% to 62.3% over the 15 months of our study[5]. Thus,

[4] These images are too small to be ads; most are 1×1 tracking pixels. We chose 50×50 since it is smaller than any of the typical online advertising format [22, 23].

[5] Our inclusion crawls failed to tag image resources for the first 3 snapshots. That is why RTB Present line in Figure 2 starts from April 2018.

Table 1: Top 10 clusters of publishers using the same ads.txt file.

#	Cluster Size	Unique Whois Servers	(Empty)	Unique Whois Registrars	(Empty)	Unique Whois Emails	(Empty)	Comments	# IPs /24	/16
1	233	19	(1)	19	(1)	12	(53)	Redirected to ads.adthrive.com/sites/UNIQ_ID/ads.txt.	156	71
2	198	23	(3)	25	(0)	13	(51)	Use ads.txt provided by MediaVine.	155	73
3	178	1	(177)	2	(176)	1	(177)	Sub-domains of livejournal.com, and use it's ads.txt.	2	2
4	106	1	(0)	1	(0)	1	(0)	Redirected to ads.iacapps.com/generic/ads.txt by MindSpark Interactive.	2	2
5	97	1	(0)	1	(0)	1	(0)	All owned by Vox Media.	7	1
6	73	6	(1)	8	(0)	4	(37)	Same website publishing platform used.	28	6
7	70	2	(68)	2	(68)	5	(6)	Sub-domains of uol.com.br.	11	7
8	56	4	(46)	12	(24)	8	(37)	Same website format (search engine). Mostly linking to izito.* and zapmeta.*.	5	4
9	56	1	(0)	1	(0)	1	(0)	Same website format (news). Same registrar and corresponding email.	4	4
10	52	16	(9)	19	(6)	16	(16)	All domains provide free video streaming (mostly for movies and porn).	48	25

although the majority of popular, ad-revenue supported publishers on the web have adopted ads.txt, there are still a significant number of websites that remain vulnerable to ad inventory fraud attacks (see § 2.2).

Alexa Rank of ads.txt Publishers. Next, we investigate how ads.txt adoption varies by publisher's popularity. Figure 3 shows the frequency count of publishers with ads.txt files binned into groups of 1,000 by Alexa rank, drawn from two snapshots taken one year apart. Although adoption is uniformly higher in April 2019 as compared to April 2018, across both snapshots we see the same trend: publishers with high Alexa ranks have higher ads.txt adoption. For example, the adoption rate is ~40% for the Top-1K publishers as compared to ~10% for the Bottom-1K in the April 2019 snapshot. If we consider only those publishers that run RTB ads, the adoption for the Top-1K (Bottom-1K) becomes 87% (3%). This is a positive, if somewhat expected trend, since popular (i.e., lucrative) publishers may be higher-value targets for ad inventory fraud attacks.

4.1.1 Correctness. Now that we have identified all publishers with ads.txt files in each snapshot, we can start analyzing the contents of these files. For a given publisher p, we validate all the records in its ads.txt file according to the IAB specification to identify syntactic errors (see § 2.4). Note that at this point, we do not attempt to validate the correctness of sellers; we defer this analysis to § 4.2.

Figure 4 shows the number of valid and invalid records in ads.txt files for all the publishers in two snapshots. Our first observation is that the size of ads.txt files grew between April 2018 and 2019: the number of valid records increased from 25 to 40 at the 50^{th} percentile over this year.[6] This occurred because existing publishers added more sellers to their files, and because new publishers with relatively long ads.txt files adopted the standard over the year-long period. Our second observation is that a minority of publishers have large ads.txt files: 33% of publishers have ads.txt files with ≥100 valid entries, and 1% have ≥1000 valid entries. Broadly speaking, there are two types of websites that fall into these ranges: (1) well-known publishers like cnn.com and espn.com that have a large, valuable impression inventory and thus maintain relationships with many ad exchanges, or (2) platforms like wordpress.com and ucoz.com that provide hosting for thousands of small, independent publishers. Our final observation from Figure 4 is that 10% of the publishers have ≥1 invalid record in their ads.txt file.

[6] This observation also matches Lukasz Olejnik's findings [69].

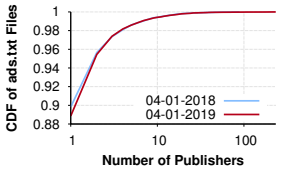

Figure 5: Number of publishers using the same ads.txt file.

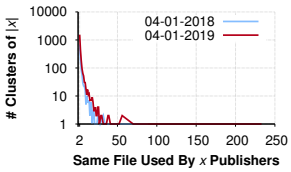

Figure 6: Clusters of $|x|$, where x is the # of publishers using the same file.

4.1.2 Clustering Publishers Using ads.txt. In theory, each publisher should have a unique ads.txt file, since they have unique IDs in each exchange marketplace (see § 2.4). However, we observed some publishers distributing identical ads.txt files.

To investigate this surprising finding we plot Figure 5, which shows the number of publishers distributing each unique ads.txt file in our dataset. We find that ~ 10% of the ads.txt files are distributed by >1 publisher, and that this fraction is invariant over time. The most common ads.txt file in our dataset was distributed by 233 publishers in the April 2019 snapshot. Figure 6 shows the number of clusters of size x, where a cluster is defined as a group of publishers distributing the same ads.txt file. For example, there is a single cluster of publishers of size 233, and 1,539 clusters of size two distributing identical files.

To gain a better understanding of why these publishers are distributing identical ads.txt files, we manually analyzed the top 10 largest clusters. For each cluster, we (1) crawled the WHOIS registry data for its constituent publishers and (2) resolved the publisher domains to IP addresses and checked how many belonged to the same /24 and /16 subnets. Additionally, we randomly sampled 20 websites from each cluster and manually inspected their homepages and ads.txt files.

The results of our investigation are shown in Table 1. For each top-10 cluster, we show the number of unique servers, registrars, and contact email addresses from WHOIS associated with publishers in that cluster, as well as the number of unique /16 and /24 IP address ranges containing the publisher's IP addresses. For most of the clusters, the WHOIS information was shared across most or all of the individual clusters, strongly suggesting that the publishers in the cluster share a common owner or at least common management. The exceptions are clusters #3, #7, and #8, where most of the WHOIS records were private (and thus labeled as "empty" in our dataset). We see similar overlap with respect to IP address prefixes

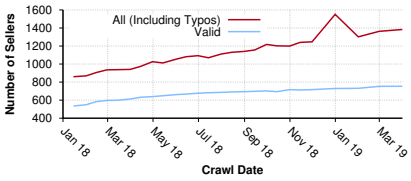

Figure 7: Number of seller domains over time.

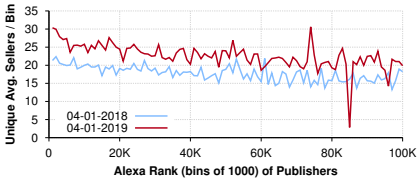

Figure 8: Authorized sellers over Alexa.

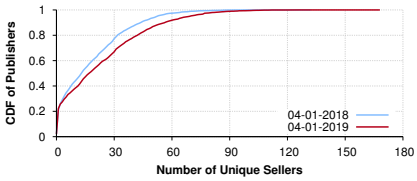

Figure 9: Sellers across two snapshots.

for clusters #3–5, #8, and #9, which is suggestive of common hosting infrastructure.

Manual investigation revealed three reasons for these large clusters of publishers. First, several clusters represent media properties with a common owner. For example, all of the publishers in cluster #5 were owned by Vox Media. Clusters #4, #8, #9, and #10 also each appear to have a single owner, respectively. Second, several clusters represented media platforms that host independent publishers, including clusters #3 (LiveJournal) and #7 (UOL). Third, several clusters represent independent publishers that happen to use consolidated SSP services. In particular, AdThrive (cluster #1) and MediaVine (#2) both appear to use their own publisher IDs when selling impression inventory, rather than having their pool of publishers all sign up for individual accounts with the ad exchanges.

4.2 Seller's Perspective

In this section, we shift perspective to focus on the sellers that are listed in `ads.txt` files. Sellers are the most important part of an `ads.txt` file, since the whole point of the standard is for publishers to authorize sellers to sell their inventory.

To perform this analysis, we must first filter out the erroneous sellers that appear in `ads.txt` files. As described in § 3.1.1, we leverage WHOIS registry data and DNS resolution to identify all the syntactically invalid seller domains. Figure 7 shows the number of unique sellers we observe in each crawl before (*All* line) and after (*Valid* line) we filter out invalid sellers. We observe that the total number of sellers increases from 860 to 1,400 over time, with the union over time containing 2,381 sellers. However, after we filter out the invalid sellers, the number of seller domains grows at a modest rate. This result is expected, since it requires significant effort for new SSPs and ad exchanges to establish themselves in the marketplace.

The union of valid sellers over time is 1,035 unique sellers, i.e., 56.4% of the seller domains in the `ads.txt` files contained syntactic errors. We focus on these seller for the remainder of our analysis. Note that this set **over-estimates** the number of valid sellers, since it may include semantically incorrect sellers. Figure 12 (discussed later) indicates that up to 20% of the unique sellers may be erroneous due to semantic errors, however these sellers only appear in a single `ads.txt` file throughout our dataset, meaning they have very limited impact on our analysis.

Sellers Per Publisher. Next, we compare the Alexa rank of publishers versus the number of sellers they authorize in their `ads.txt` files. Figure 8 presents the average number of valid sellers across bins of 1000 publishers sorted by their Alexa rank, with separate lines for our April 2018 and 2019 snapshots. We see that the average number of sellers at every rank has grown over the year: there were ~ 10 more sellers per bin in the April 2019 snapshot as compared to April 2018. This is primarily due to publishers forming new partnerships with existing sellers, rather than the emergence of new sellers over time (see Figure 7). Additionally, we find that publishers at higher ranks have listed more authorized sellers on average, possibly because their impression inventory is more valuable, thus making them more desirable partners to ad exchanges.

Figure 9 shows the number of unique sellers listed within each publisher's `ads.txt` file for two snapshots of our crawl. We make three observations: first, ~2% of the publishers have no sellers in their files. We manually examined these `ads.txt` files and found that they were either empty or just contained comments (e.g., `https://www.youtube.com/ads.txt`). These empty `ads.txt` files are intentionally installed by publishers, since they signal to ad exchanges and DSPs that **nobody** is authorized to sell their impressions. Second, the median publisher listed 17 sellers in their `ads.txt`, while the top 20% of publishers listed ≥42 unique sellers in their `ads.txt`'s. Finally, we see that the number of unique sellers per publisher has increased slightly year-over-year, with the increases mostly concentrated amongst the publishers with the largest `ads.txt` files.

Table 2 focuses on the top 20 publishers who have listed the most unique sellers in their `ads.txt` files.[7] One interesting observation is that there is no correlation between Alexa rank and unique sellers for the top 20 publishers. They do have a common theme though — they are all news websites. Another notable observation is the difference between the number of unique sellers and number of valid entries per publisher. The latter is an order of magnitude greater than the former because a publisher can have multiple publisher IDs associated with a given seller (see § 2.4). This is highlighted in Figure 10, which compares the count of unique sellers, total publisher IDs, and unique publisher IDs per publisher for `ads.txt` files in our April 2019 snapshot. We see an order of magnitude more publisher IDs than unique sellers. This conclusion remains the same even if we de-duplicate publisher IDs, which makes sense because duplicate publisher IDs within a given `ads.txt` file would be errors.

Recall that each publisher ID associated with a seller also has a specific relationship with the seller. This relationship can be of two types: *Direct* or *Reseller* (see § 2.4). For example, as shown in Table 2, `arcmax.com` has 3,617 publisher IDs for 168 unique sellers.

[7] Others have also observed that sites like `arcamax.com` and `breitbart.com` have unusually large `ads.txt` files [69, 90].

Table 2: Top 20 publishers with most sellers. Direct and Reseller are their seller account relationships.

Publisher	Alexa Rank	# Unique Sellers	Valid Entries	Relationship D	R
arcamax.com	22565	168	3617	434	3183
breitbart.com	242	158	980	123	857
walterfootball.com	48279	148	2805	394	2411
investing.com	408	130	1551	218	1333
webconsultas.com	13730	127	2309	263	2046
shoppinglifestyle.com	72547	119	1249	155	1094
moretvtime.com	17380	118	2408	231	2177
newindianexpress.com	13028	118	1967	225	1742
americanlisted.com	53358	117	1239	146	1093
thehindu.com	1067	117	1210	127	1083
thegatewaypundit.com	8429	116	1501	217	1284
vikatan.com	6005	114	1046	168	878
flvto.biz	889	114	3490	289	3201
realgm.com	11118	112	1397	186	1211
fayerwayer.com	18578	111	1944	12	1932
publimetro.co	40324	111	1944	12	1932
pjmedia.com	16437	111	1522	140	1382
metroecuador.com.ec	27378	111	1944	12	1932
nuevamujer.com	40645	111	1944	12	1932
publimetro.com.mx	21623	111	1944	12	1932

Table 3: Top 20 sellers. Publishers have either Direct, Reseller, or Both relationships with them.

Authorized Seller	# of Publishers	Relationship D	R	B	Avg. (Median) Entries / Publisher	
google.com	17771	5305	1408	11058	14.39	(4.00)
appnexus.com	12825	578	5127	7120	15.24	(8.00)
rubiconproject.com	12691	1145	4969	6577	8.35	(5.00)
openx.com	12250	652	5432	6166	13.04	(7.00)
pubmatic.com	12112	605	6345	5162	13.80	(7.00)
indexexchange.com	11347	977	4713	5657	6.22	(4.00)
contextweb.com	10405	275	7214	2916	7.97	(4.00)
spotxchange.com	10197	292	7046	2859	7.16	(4.00)
spotx.tv	9957	299	7009	2649	6.64	(4.00)
advertising.com	9819	310	6705	2804	7.48	(4.00)
sovrn.com	9146	1612	3925	3609	3.97	(2.00)
adtech.com	9110	1103	4803	3204	4.61	(3.00)
freewheel.tv	9029	170	6729	2130	23.52	(7.00)
tremorhub.com	8529	260	6955	1314	5.32	(3.00)
smartadserver.com	8401	441	5836	2124	5.67	(3.00)
districtm.io	7599	1730	2015	3854	3.23	(2.00)
lkqd.net	7300	54	5589	1657	4.78	(3.00)
aolcloud.net	7298	855	4732	1711	3.31	(2.00)
lijit.com	7100	2236	2210	2654	3.11	(2.00)
teads.tv	6757	3406	1976	1375	2.49	(2.00)

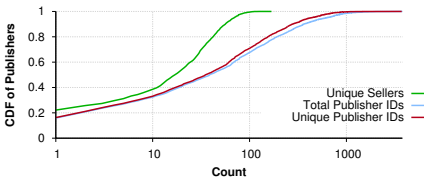

Figure 10: Number of sellers and associated publisher IDs (April 2019).

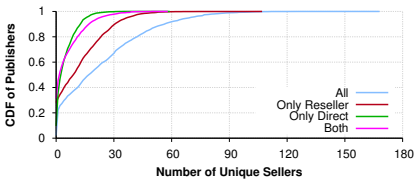

Figure 11: Sellers by publisher relationships (April 2019).

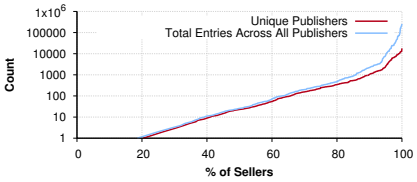

Figure 12: Number of unique publishers and total entries for sellers.

Out of these 3,617 IDs, 434 have a *Direct* relationship, meaning the publisher directly controls the given account. For the remaining 3,183 *Reseller* IDs, the publisher has authorized another entity to control this account associated with the seller.

Figure 11 breaks down the valid entries in each publishers' `ads.txt` files by relationship type for our April 2019 snapshot. The *All* line is identical to Figure 9, and is shown here for scale. The *Only* lines count cases where a publisher only has a *Direct* or *Reseller* relationship (respectively) with a seller, while the *Both* line counts cases where the publisher has both relationships with a given seller. Overall, we see that *Reseller* relationships are most common: 25% of the publishers have only *Reseller* relationships with ≥20 sellers, whereas just 2% of the publishers have only *Direct* relationship with ≥20 sellers. The *Both* line is almost coincident with the *Only Direct* line, suggesting that when a publisher has a *Direct* relationship with a seller, they almost always have a *Reseller* relationship with that seller as well.

Seller Popularity. So far, we have looked at authorized sellers with respect to each publisher. Now we look at the popularity of sellers across all publishers in our dataset.

Figure 12 shows each sellers' popularity in terms of (1) the total number of entries they appear in across all publishers, and (2) the number of unique publishers they have relationships with. We observe that 20% of the sellers are only involved with a single publisher. Some of these sellers are semantic errors (e.g., `googlesyndication.com` instead of `google.com`), some are typos (e.g., `comgoogle.com`), and some are legitimate ad *networks* (not exchanges, e.g., `zergnet.com`) that have been added to the `ads.txt` file by mistake (see § 3.1.1). At the other extreme, the top 25% and top 10% of sellers are listed on ≥250 and ≥1050 publishers, respectively. This result is expected, since there are powerful network effects that draw publishers to the biggest ad exchange markets. Lastly, the top sellers have an order of magnitude more entries in comparison to their publisher presence. This bolsters our finding that publishers tend to register multiple accounts with top sellers.

Table 3 shows the top 20 sellers listed in the `ads.txt` files in our dataset. Unsurprisingly, the top ad companies like Google, OpenX, and Rubicon are present in this list. `google.com` is the most popular seller, and is associated with 17.7K publishers. Furthermore, it appears in 14.4 entries per `ads.txt` file on average. From the table, we can see that publishers tend to have both direct and reseller relationships with the top sellers.

5 COMPLIANCE WITH ADS.TXT

In § 4, we looked at how Alexa Top-100K publishers have adopted the `ads.txt` standard over the course of 15 months, and which

ad sellers they have authorized to sell their inventory during RTB auctions. In this section, we take the next step and try to examine the ads.txt standard from the ad buyers' side. After all, one of the major goals of ads.txt is to enable ad buyers (e.g., DSPs) to verify the authenticity of inventory before bidding. Thus, we pose the question: *are buyers complying with the* ads.txt *standard by only purchasing impression inventory via authorized sellers?*

5.1 Isolating RTB Ads

To determine whether ad buyers are complying with the ads.txt file for a given publisher p, we first need to identify ads which were served through RTB auctions on p. This is important, since ads.txt compliance only matters for RTB auctions.

Using our methodology from § 3.2.1, we extract all inclusion chains rooted in p. Then, as described in § 3.2.2, we use EasyList to identify all chains that eventually serve an ad on p. From these *ad inclusion chains*, we can further isolate just the ads served via RTB using two insights. First, we know that for an ad to be served via RTB, there must be at least 3 parties involved: the publisher, the exchange (seller), and the DSP (buyer). Thus, we filter out all the ad inclusion chains with < 3 resources. Second, through the ads.txt dataset, we have a lower-bound estimate on all the ad exchanges (sellers) used by Alexa Top-100K publishers (set S_v, see § 3.1.1). Using these 1,035 sellers, we filter out all ad inclusion chains that have zero resources from the set of valid sellers.

After applying all the filters above, we are left with 135M RTB ad inclusion chains. Although we cannot claim that these chains capture all of the ads in our dataset served by RTB, they should cover the ads served by authorized sellers listed in ads.txt files.

5.2 Compliance Verification Metrics

Now that we have isolated the inclusion chains that served RTB ads, we can investigate compliance with the ads.txt standard by ad buyers. To accomplish this, must first carefully process our dataset using the following set up steps.

Seller–Buyer Pairs. *First*, we create a set R_p of seller–buyer tuples (s, b) for each publisher p. s and b are derived from RTB ad inclusion chains, such that s and b are the 2^{nd}-level domains of the chain elements at index i and $i+1$ respectively. For example, consider an ad inclusion chain $p \rightarrow e_1 \rightarrow e_2 \rightarrow e_3 \rightarrow d$, rooted at publisher p. The last element of the chain d is the DSP that ultimately served the ad. e_1, e_2 are both exchanges, and are present in the set of valid authorized sellers S_v, whereas $e_3 \notin S_v$. In this case, we would produce the buyer–seller tuples (e_1, e_2) and (e_2, e_3), since e_2 bought and then resold the impression. Lastly, note that since we only include tuples where s is a member of the ads.txt authorized sellers set S_v, we do not consider the tuple (e_3, d) in R_p.

Non-Compliant Pairs. *Second*, we derive the set of non-compliant (s, b) tuples $R_p^\star$ for p, such that $s \notin S_p$, where S_p is the set of authorized sellers listed by p in its ads.txt file. Intuitively, the tuples in $R_p^\star$ capture cases where a seller was not authorized by the publisher to sell its inventory.

Clustering Domains. *Third*, we clustered domains together that belong to the same organization. This step is necessary because of a quirk of the ads.txt data: recall from § 2.4 that the seller domains listed in field #1 of ads.txt files are not necessarily the domains that host ad auctions. For example, Google specifies that its seller domain is google.com, even though the actual auctions are hosted at doubleclick.net. These discrepancies in S_p can lead to incorrect compliance analysis if they are not addressed. For example, say that google.com $\in S_p$ for publisher p. If we observe an ad buyer b purchasing ad impressions from doubleclick.net during RTB auctions, we would incorrectly mark doubleclick.net and b as the non-compliant seller and buyer respectively.

To address this issue, we clustered domains together that belong to the same organization using data provided by *WhoTracksMe* [95]. This dataset is gathered by *Cliqz*, which is a German company that develops a privacy-preserving web browser and extensions [21].[8] This dataset contains mappings for 28 parent domains, including Google, OpenX, Rubicon Project, etc. Using this dataset, we map the domains that appear in our RTB ad inclusion chains and the domains from S_v to their parent domain.

Filter Self-edges. *Fourth*, after clustering we filter out all tuples (s, b) where $s = b$. Such edges are common in our data, and represent instances where an ad exchange redirected the browser to back to themselves. This may occur because the ad exchange decided to purchase the impression themselves, or for internal bookkeeping purposes. Regardless, transitions from s to s are irrelevant with respect to measuring compliance with ads.txt.

Measuring Compliance. Finally, using $R_p^\star$, we calculate *unweighted compliance* for p as the percentage of non-compliant tuples over the total tuples $100 * |R_p^\star|/|R_p|$. However, this metric is not necessarily fair, since it does not take into account the relative frequency that sellers–buyer pairs appear in the ad inclusion chains. To account for frequency, we also calculate *weighted compliance* as $\sum_{\forall i \in R_p^\star} f(i) / \sum_{\forall j \in R_p} f(j)$, where $f(t)$ is the number of times tuple t appears in RTB ad inclusion chains on p.

5.3 Results

Figure 13 show the percentage of non-compliant tuples per publisher in our April 2019 snapshot. We see that the percentage of publishers whose inventory is filled under total compliance is more than 70% in both the weighted and non-weighted cases. Compliance for weighted cases is substantially higher than that of non-weighted case due to the fact that a small number of compliant exchanges (e.g., DoubleClick) auction a disproportionaty large amount of inventory. Overall, we can conclude that the vast majority of RTB ads in our dataset appear to have been served by buyers who were in compliance with publishers' ads.txt files. This is an encouraging result as it demonstrates that publishers are willing to adopt standards that can counter fraud and bring transparency to the opaque RTB ecosystem.

Distance. One interesting question is *when do non-compliant ad auctions occur in the inclusion chains?*, i.e., in the seller that directly receives the impression from the publisher, or farther down the chain? Figure 14 shows the average distance of the buyer from the very first authorized seller for complaint and non-compliant

[8] We provide the list of clustered domains along with their parent domains in our open-sourced dataset.

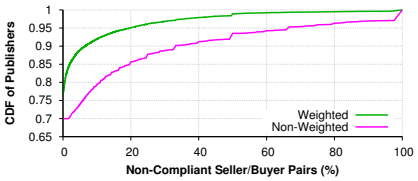

Figure 13: Percentage of non-compliant seller–buyer tuples per publisher. Domains are clustered by their parent domain.

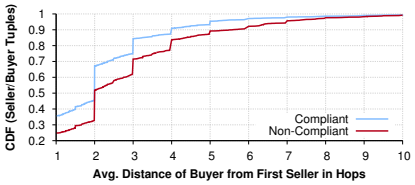

Figure 14: Average distance of buyer from first seller across all publishers. Distances are shown for both compliant and non-compliant tuples.

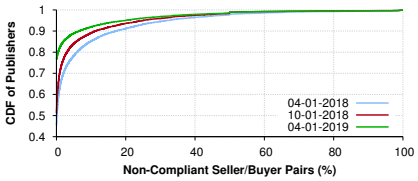

Figure 15: Percentage of non-compliant seller–buyer tuples per publisher over time. Results are shown for the weighted tuples.

Table 4: Top 20 non-compliant Seller-Buyer pairs, sorted by presence on number of unique publishers.

Seller	Buyer	# Publishers (%)		Total Chains (%)	
gumgum	domdex	247	20.38	280	16.25
gumgum	appnexus	225	20.49	237	20.10
taboola	weborama	188	52.66	190	51.77
taboola	rubiconproject	154	11.55	404	11.31
dailymotion	dyntrk	148	51.21	1296	42.99
taboola	indexexchange	144	11.61	190	11.59
gumgum	pubmatic	139	27.25	480	28.27
justpremium	openx	138	100.00	936	100.00
criteo	media	120	74.53	454	77.47
rubiconproject	yahoo	120	2.63	120	2.63
criteo	yieldlab	105	78.36	756	80.51
taboola	pubmatic	104	12.87	512	12.41
springserve	pubmatic	103	49.28	4668	53.84
exponential	google	101	31.46	1700	20.83
criteo	ligadx	98	77.78	502	83.11
criteo	pubmatic	84	82.35	415	78.60
nativeroll	weborama	81	100.00	647	100.00
nativeroll	seedr	78	100.00	464	100.00
aniview	google	76	84.44	5047	82.21
yandex	google	65	98.48	1744	97.76

Table 5: Percentage of `ads.txt`-enabled publishers on top sellers.

Seller	% Publishers w/ `ads.txt`	# Publishers w/ RTB Ads
google	58.64	23552
advertising	75.46	7196
pubmatic	79.53	6800
rubiconproject	88.37	5562
openx	91.18	3173
appnexus	91.71	3150
sovrn	90.61	2279
indexexchange	88.98	1915
teads	93.99	1232
smartadserver	92.17	1085

tuples. We observe a clear separation between the lines, with non-compliant buyers tending to be one hop farther away from the first seller than complaint buyers on average. This confirms our intuition that compliance with the `ads.txt` standard tends to be stronger earlier in chains, when top sellers are typically conducting the auctions. In contrast, as the chain length grows, less reputable buyers and sellers become involved, and compliance wanes.

Non-Compliant Sellers. Next, we take a deeper look into the seller and buyer domains from the non-compliant tuples. Table 4 shows the top 20 non-compliant tuples across all publishers, after clustering them by their parent domains. For each tuple, we show the total number and percentage of publishers it was non-compliant on. Table 4 also shows the total number and percentage of times the tuple was non-compliant across all publishers.

With respect to the non-compliant sellers, several companies appear to be systematically non-compliant, such as NativeRoll, GumGum, Criteo, and JustPremium. Only one of the top authorized sellers from Table 3 (Rubicon Project) appears on the list. However, it is only non-compliant with a single buyer and only in 2.6% of transactions in our dataset. This finding suggests that top authorized sellers like Google and OpenX are enforcing compliance with the `ads.txt` standard within their markets.

One possibility is that top sellers are only auctioning impression inventory that can be validated, i.e., from publishers with `ads.txt` files. However, this is not the case: Table 5 shows (1) the number of publishers in our dataset that had RTB ad inclusion chains with the given seller, and (2) the percentage of these publishers that had `ads.txt` files. For example, only 59% of the publishers in our dataset whose impression inventory moved through Google's exchange had an `ads.txt` file. This demonstrates that all of the top sellers are, to some extent, still auctioning inventory that cannot be validated using `ads.txt`.

A second possibility is that top sellers are faithfully following the `ads.txt` standard by refusing to auction unauthorized impressions. Although our data suggests that this might be the case, we cannot guarantee this from observational data alone. We attempted to become a publisher in order to conduct controlled experiments to test compliance with the `ads.txt` standard, but we were unable to do so.[9]

Non-Compliant Buyers. With respect to non-compliant buyers, the striking feature of Table 4 is that most are actually SSPs/ad exchanges, including eight of the top authorized sellers from Table 3. In other words, top DSPs seem to be following the `ads.txt` standard by not buying non-compliant inventory. Rather, *sellers* are buying non-compliant inventory, although the reason for this is unclear, since it seems unlikely that they are able to resell this non-compliant inventory at auction. Many of these companies offer

[9] All of the ad exchanges we contacted refused to engage with us unless our website received on the order of millions of unique visitors per month.

seller and buyer-side products, so it is possible that they are purchasing this non-compliant inventory and then serving ads, rather than reselling. Still, this behavior is surprising given that many of these companies have called for strict enforcement of the ads.txt standard [7, 40, 71, 81].

Compliance Over Time. Finally, we examine compliance with the ads.txt standard over time. Figure 15 shows the non-compliant, weighted tuples for three snapshots roughly five months apart. We can see that the percentage of compliant inventory sales has been steadily increasing over time. The percentage of completely compliant publishers rose from 46% in April 2018 to 77% in April 2019. Again, this is an encouraging result for the ads.txt standard: we have observed not only a healthy adoption of the standard, but also an improvement in compliance over time.

6 RELATED WORK

In this section, we survey related work on the online advertising ecosystem. We also discuss studies on the topic of cookie matching and transparency tools. Next, we discuss related work on the ecosystem of ad fraud and prevention mechanisms. Finally, we conclude with related work on the ads.txt standard.

6.1 The Online Advertising Ecosystem

Researchers have been studying online advertising ecosystem for almost a decade. Mayer et al. presents an overview of this topic in [62]. Barford et al. mapped the online *adscape* through targeted ads by major ad networks on the web [11], whereas Rodriguez et al. [93] and Razaghpanah et al. [79] measured the ad ecosystem on mobile devices. Using browsing traces, Gill et al. demonstrated that advertising revenue is skewed towards large companies like Google [38]. Guha et al. [44] and Carrascosa et al. [17] developed controlled methodologies to study individual implications of targeted advertising. Researchers have also found evidence of advertisers using sensitive attributes to target users [26, 94, 96]. Studies have also highlighted ads being served for malicious purposes [89, 98], and through covert channels [13].

Tracking. Advertising companies track users around the web to build profiles about them, so that later they can serve targeted ads to users. Krishnamurthy et al. were the first to document the pervasiveness of online tracking [53–55]. Lerner et al. provided a longitudinal measurement of third-party tracking from 1996 to 2016 [59]. More recently, Cahn et al. and Englehardt et al. conducted large scale crawls on Alexa Top-10K and Alexa Top-1M to provide an in-depth analysis of web tracking [16, 34]. Falahrastegar et al. looked at tracker prevalence across geographic regions [35].

RTB and Cookie Matching. More recently, the online ad ecosystem has shifted towards RTB [97]. Through cookie matching, which is a pre-requisite for RTB, advertisers exchange user identifiers with each other. Acar et al. conducted crawls on Alexa Top-3K and found that hundreds of domains passed unique identifiers to each other [1]. Olejnik et al. discovered 125 cookie matching ad exchanges by studying winning bid prices during RTB auctions [70]. Falahrastegar et al. used crowd-sourced browsing data to identify domains sharing unique identifiers [36]. Bashir et al. used retargeted ads to examine cookie matching [12]. They further conducted simulations to highlight the extent of information sharing by ad exchanges behind the scences [15]. By collecting winning prices from the network traffic, Olejnik et al. [70] and Papadopoulos et al. [72] examined how much advertisers are paying for users in RTB auctions.

Transparency. Research surveys have shown that users have grown increasingly concerned about the state of online tracking [10, 63]. Users feel that they don't have meaningful choice in how their data is collected by advertisers [5, 61, 92]. Similarly, Leon et al. found lack of control over data sharing as a major cause for users' unwillingness to share information with advertisers [58]. In their user surveys, Dolin et al. found that users were more comfortable with targeted ads when they were given explanation on how a targeted ad was served [30]. These studies suggest that users feel there is a lack of transparency in the advertising ecosystem.

In an effort to make the advertising ecosystem more transparent, some advertising companies (e.g., Google, Facebook) have built transparency tools called Ad Preference Managers (APMs) to enable users see what information has been inferred about them. However, studies have highlighted certain issues with these tools: they lack coverage [6, 96], exclude sensitive user attributes [26], and infer noisy and irrelevant interests [14, 29, 91].

6.2 Ad Fraud

Over the years, numerous white-papers and blog posts have been published by researchers and advertisers, documenting the issues pertaining to ad fraud. In 2016, the IAB published a white-paper highlighting that ad fraud costs advertisers $8.2B per year [49, 84]. Similarly, the Association of National Advertisers (ANA) reported ad fraud costs of $7.2B in 2016 [86]. Daswani et al. present an accessible introduction to the topic of ad fraud in [24].

Researchers have proposed methodologies to study various forms of ad fraud. Springborn et al. examined the extent of impression fraud by setting up honeypot websites [87]. Dave et al. provided a systematic look at click-spam, and proposed an automated methodology to fingerprint click-spam attacks [27]. Some studies have provided case studies on botnets conducting click-spam [25, 67, 73]. Haddadi et al. [45] used bluff ads to detect click fraud. Stone-Gross et al. studied ad fraud in ad exchanges [88].

Several prevention mechanisms have also been introduced in the literature. Zhang et al. and Metwally et al. proposed methodologies to combat ad fraud by identifying duplicate clicks [65, 99]. Metwally et al. further proposed an approach to detect click fraud by looking for similarities among fraudsters [66]. Nazerzadeh et al. provided an approach based on economic incentives to counter ad fraud [68]. However, sophisticated botnets like *ZeroAccess* [85] and *ClickBot.A* [20] can evade such prevention mechanisms. Pearce et al. and Daswani et al. outlined techniques to combat fraud from botnets [25, 73]. WhiteOps published a report on their take down of the infamous *Methbot* [64].

Domain spoofing has been a major issue in programmatic advertising. A good introduction to domain spoofing is provided in [50, 51]. Recently, *Methbot* spoofed domains for more than 6,000 premium publishers to generate revenue of $5M per day [18]. In November 2017, Adform published a white-paper describing how they

took down *HyphBot*, which was generating 1.5B spoofed requests per day [47].

6.3 ads.txt Adoption

Besides a white-paper and some blog posts, to the best of our knowledge, there is no prior work which provides an in-depth, longitudinal analysis of the ads.txt standard.

Lukasz Olejnik, an independent researcher, recently published a white-paper on his longitudinal study of the ads.txt standard [69]. Olejnik conducted gathered ads.txt data on Alexa Top-100K publishers from August 2017, right after the inception of the ads.txt, to March 2018. He performed one more crawl towards the end of December 2018. Results from this white-paper corroborates our findings regarding longitudinal trends in adoption and top sellers. Olejnik did **not** study the compliance aspect of the standard.

Since the inception of ads.txt standard, several blog posts have studied its trends, and different companies have reported different trends. Pixalate reported a x5 growth in ads.txt adoption in 2018, with 75% of the top 1,000 programmatic domains adopting the standard[75]. They also claim that ads.txt has reduced ad fraud by 10% [76]. According to OpenX, 60% of the top 1,000 publishers (comScore's list) have adopted the standard [74]. First Impressions' reported adoption trends on Alexa Top-1000 sites are similar to ours [37]. Some blogs also noticed errors in publishers' ads.txt files [37, 74].

Several companies, including Google, provide tools for publishers to generate and validate their ads.txt records [2, 4, 39].

In their bid to eliminate the ability to profit from counterfeit inventory and bring more transparency to programmatic advertising, IAB has recently introduced ads.txt like standard for mobile apps, called app-ads.txt [42]. Furthermore, IAB is working towards introducing another standard called sellers.json, which will allow the buyers to discover the identities of all the authorized reseller partners of a participating seller (SSP) [43].

7 LIMITATIONS

In this section, we describe the limitations that should be considered for the results presented in this study.

First, we rely on EasyList [32] to detect inclusion chains that end up serving advertisements. These lists are manually curated over time and may introduce errors. For example, if we classify a benign (advertisement) chain as advertisement (benign), we may end up over-estimating (under-estimating) non-compliance if the seller was not listed in the publisher's ads.txt file. Additionally, we do not use any supplementary, language-specific filter lists to identify advertisements on non-English websites.

The ideal way to check for non-compliance would be to become part of the ecosystem as a publisher and serve ads. As a publisher, we could control the contents of our ads.txt file and monitor tags that serve advertisements. However, this approach is quite challenging to implement: it requires us to form relationships with popular ad exchanges, and the top exchanges do not form partnerships unless your website has millions of unique visitors per month.

Second, the clustering process in § 5.2 is not perfect. We manually mapped 101 domains to 28 parent domains using the data from WhoTracksMe [95]. Although we made sure that we clustered popular domains by going through the list of top 30 seller–buyer tuples, we could have missed some domains that should have been clustered.

Finally, this study does not analyze the ads.txt standard on the mobile ecosystem. In March 2019, the IAB introduced the ads.txt standard for the mobile apps, called app-ads.txt [42]. A separate study is required to understand the adoption of this standard and compliance in the mobile ecosystem.

8 CONCLUDING DISCUSSION

In this study, we present the first large-scale, longitudinal study of the ads.txt standard. Using data crawled from 240K websites over a period of 15 months, we examine the adoption of ads.txt by publishers, the contents of these files, the characteristics of sellers who appear in the files, and compliance with the standard by sellers and buyers.

Compliance. One of the motivating questions behind our study was *are members of the online ad ecosystem complying with the* ads.txt *standard?* The answer to this question is: somewhat. With respect to adoption, we found that over 60% of popular publishers that are monetized via RTB ads have adopted ads.txt, which is impressive for a standard that is just over two years old (as of this writing). Further, our analysis of ad inclusion chains strongly suggests that SSPs and ad exchanges are honoring the standard by not attempting to sell unauthorized inventory. Future work should attempt to validate this using causal experiments.

That said, there is a great deal of room for improvement before domain spoofing will be eradicated. There are still many publishers that have not adopted ads.txt, and their impression inventory continues to be purchased from SSPs/ad exchanges. All of these domains are vulnerable to spoofing. Additionally, we do observe specific sellers that continue to sell impressions that they are not authorized to sell, as well as specific buyers (including many top ad exchanges) who continue to purchase impressions from these unauthorized sellers. All of these companies run the risk of introducing spoofed inventory into the marketplace.

Transparency. The other motivating question of our study was *how useful is* ads.txt *as a transparency mechanism?* Here again, the answer is mixed. On the positive side, ads.txt is enjoying wide adoption. For the first time ever, publishers are explicitly declaring who they have advertising contracts with. Further, by aggregating across ads.txt files, it is possible to compile an explicit and extensive list of seller-side advertising platforms. Coupled with inclusion chain data, buyer-side platforms can also be identified. These datasets are extremely useful for measurement studies of the online ad ecosystem, which historically have had to rely on heuristics or crowdsourced data (e.g., EasyList) to identify these domains. Additionally, this data may be useful for browser extensions that inform users about the advertising practices of publishers [69] or block ads.

However, there are several caveats to the ads.txt data. First, as we saw throughout our study, ads.txt files contain various classes of errors that must be mitigated by consumers of the data. Fortunately, we develop techniques in this study that can help in this regard. Second, ads.txt is only designed to make advertising domains transparent, not tracking domains. Additional datasets

and detection techniques are still necessary to identify trackers. Finally, we note that the seller domains listed in ads.txt files are not all-inclusive; additional, manual work is required to map seller domains like google.com to all of other domains used by sellers.

Future Directions. The results from this study can be used by both privacy researchers and stakeholders in the advertising ecosystem. Privacy researchers have been long trying to understand the roles (e.g., tracker, advertiser, ad exchange, etc.) of third-parties participating in the ecosystem [12, 15]. We demonstrate in this study that it is possible to compile an explicit and extensive list of ad exchanges. Similar studies can be conducted leveraging upcoming standards to identify buyer-side relationships. For example, the IAB is introducing another standard called sellers.json [43], under which the seller (SSP) discloses all other entities it has selling relationships with.

Organizations like the IAB can use results from this study to improve future standards. For example, we find that although ads.txt adoption is quite encouraging, publishers make mistakes in their published ads.txt files, including typos and listing non-exchanges like ad networks. Although file format verifiers are available for ads.txt [3, 9], these tools could be improved to identify non-syntax related errors. Furthermore, to account for discrepancies where the seller domain is different from the domain that hosts ad auction (e.g., google.com versus doubleclick.net), the IAB should compile and maintain a canonical list of seller domains for ads.txt. This list could also be incorporated into ads.txt file verification tools.

ACKNOWLEDGMENTS

We thank our shepherd, Georgios Smaragdakis, and the anonymous reviewers for their helpful comments. This research was supported in part by NSF grants CNS-1703454 and IIS-1553088. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

REFERENCES

[1] Gunes Acar, Christian Eubank, Steven Englehardt, Marc Juarez, Arvind Narayanan, and Claudia Diaz. 2014. The Web Never Forgets: Persistent Tracking Mechanisms in the Wild. In *Proc. of CCS*.
[2] Ads.txt Guru Manager 2018. Simplify your ads.txt management. Ads.txt Guru. https://adstxt.guru/publishers/.
[3] Ads.txt Guru Validator [n. d.]. ads.txt Validator. Ads.txt Guru. https://adstxt.guru/validator/.
[4] Ads.txt Manager [n. d.]. Ads.txt Manager | Free | Easily Manage Ads.txt Files. Ads.txt Manager. https://www.adstxtmanager.com.
[5] Lalit Agarwal, Nisheeth Shrivastava, Sharad Jaiswal, and Saurabh Panjwani. 2013. Do Not Embarrass: Re-examining User Concerns for Online Tracking and Advertising. In *Proc. of the Workshop on Usable Security*.
[6] Athanasios Andreou, Giridhari Venkatadri, Oana Goga, Krishna P. Gummadi, Patrick Loiseau, and Alan Mislove. 2018. Investigating Ad Transparency Mechanisms in Social Media: A Case Study of Facebook's Explanations. In *Proc of NDSS*.
[7] AppNexus Enforcement 2018. AppNexus Enforces Ads.txt in Broader Push for Industry Transparency. AppNexus. https://www.appnexus.com/company/pressroom/appnexus-enforces-adstxt-in-broader-push-for-industry-transparency.
[8] Sajjad Arshad, Amin Kharraz, and William Robertson. 2016. Include Me Out: In-Browser Detection of Malicious Third-Party Content Inclusions. In *Proc. of Intl. Conf. on Financial Cryptography*.
[9] Automated Validator [n. d.]. Automated: adstxt validator. Automated. https://verifyadstxt.com/.
[10] Rebecca Balebako, Pedro G. Leon, Richard Shay, Blase Ur, Yang Wang, and Lorrie Faith Cranor. 2012. Measuring the Effectiveness of Privacy Tools for Limiting Behavioral Advertising. In *Proc. of W2SP*.
[11] Paul Barford, Igor Canadi, Darja Krushevskaja, Qiang Ma, and S. Muthukrishnan. 2014. Adscape: Harvesting and Analyzing Online Display Ads. In *Proc. of WWW*.
[12] Muhammad Ahmad Bashir, Sajjad Arshad, , William Robertson, and Christo Wilson. 2016. Tracing Information Flows Between Ad Exchanges Using Retargeted Ads. In *Proc. of USENIX Security Symposium*.
[13] Muhammad Ahmad Bashir, Sajjad Arshad, Engin Kirda, William Robertson, and Christo Wilson. 2018. How Tracking Companies Circumvented Ad Blockers Using WebSockets. In *Proc. of IMC*.
[14] Muhammad Ahmad Bashir, Umar Farooq, Maryam Shahid, Muhammad Fareed Zaffar, and Christo Wilson. 2019. Quantity vs. Quality: Evaluating User Interest Profiles Using Ad Preference Managers. In *Proc of NDSS*.
[15] Muhammad Ahmad Bashir and Christo Wilson. 2018. Diffusion of User Tracking Data in the Online Advertising Ecosystem. In *Proc. of PETS*.
[16] Aaron Cahn, Scott Alfeld, Paul Barford, and S. Muthukrishnan. 2016. An Empirical Study of Web Cookies. In *Proc. of WWW*.
[17] Juan Miguel Carrascosa, Jakub Mikians, Ruben Cuevas, Vijay Erramilli, and Nikolaos Laoutaris. 2015. I Always Feel Like Somebody's Watching Me: Measuring Online Behavioural Advertising. In *Proc. of ACM CoNEXT*.
[18] Yuyu Chen. 2017. Domain spoofing remains a huge threat to programmatic. Digiday. https://digiday.com/marketing/domain-spoofing-remains-an-ad-fraud-problem/.
[19] Chrome Debugging Protocol [n. d.]. Chrome DevTools Protocol Viewer. GitHub. https://developer.chrome.com/devtools/docs/debugger-protocol.
[20] Clickbot.A 2016. Clickbot.A User Agent String. Distil Networks). https://www.distilnetworks.com/bot-directory/bot/clickbot-a/.
[21] Cliqz [n. d.]. Cliqz - The no-compromise browser. Cliqz GmbH. https://cliqz.com/en/.
[22] Common Ad Dimensions 2008. Standard Banner Sizes List. Bannersnack Blog. https://blog.bannersnack.com/banner-standard-sizes/.
[23] Common Ad Dimensions (Google AdSense) [n. d.]. Guide to ad sizes. Google. https://support.google.com/adsense/answer/6002621?hl=en.
[24] Neil Daswani, Chris Mysen, Vinay Rao, and Stephen Weis. 2008. Online Advertising Fraud. *Crimeware Underst. New Attacks Defenses* 40 (01 2008).
[25] Neil Daswani, The Google Click Quality, Security Teams, and Google Inc. 2007. The anatomy of clickbot.a. In *USENIX Hotbots*.
[26] Amit Datta, Michael Carl Tschantz, and Anupam Datta. 2015. Automated Experiments on Ad Privacy Settings: A Tale of Opacity, Choice, and Discrimination. In *Proc. of PETS*.
[27] Vacha Dave, Saikat Guha, and Yin Zhang. 2012. Measuring and Fingerprinting Click-Spam in Ad Networks. In *Proc. of SIGCOMM*.
[28] Jessica Davies. 2017. WTF is ads.cert? Digiday. https://digiday.com/media/what-is-ads-cert/.
[29] Martin Degeling and Jan Nierhoff. 2018. Tracking and Tricking a Profiler: Automated Measuring and Influencing of Bluekai's Interest Profiling. In *Proc. of WPES*.
[30] Claire Dolin, Ben Weinshel, Shawn Shan, Chang Min Hahn, Euirim Choi, Michelle L. Mazurek, and Blase Ur. 2018. Unpacking Perceptions of Data-Driven Inferences Underlying Online Targeting and Personalization.
[31] DoubleVerify. 2019. DoubleVerify Fraud Lab Identifies Botnet Scheme Targeting Ads.txt. DoubleVerify. https://www.doubleverify.com/newsroom/doubleverify-fraud-lab-identifies-botnet-scheme-targeting-ads-txt/.
[32] EasyList [n. d.]. EasyList. The EasyList authors.. https://easylist.to.
[33] eMarketer Programmatic Ad Spending 2018. US Programmatic Ad Spending Forecast Update 2018. eMarketer. https://www.emarketer.com/content/us-programmatic-ad-spending-forecast-update-2018.
[34] Steven Englehardt and Arvind Narayanan. 2016. Online tracking: A 1-million-site measurement and analysis. In *Proc. of CCS*.
[35] Marjan Falahrastegar, Hamed Haddadi, Steve Uhlig, and Richard Mortier. 2014. The Rise of Panopticons: Examining Region-Specific Third-Party Web Tracking. In *Proc of. Traffic Monitoring and Analysis*.
[36] Marjan Falahrastegar, Hamed Haddadi, Steve Uhlig, and Richard Mortier. 2016. Tracking Personal Identifiers Across the Web. In *Proc. of PAM*.
[37] First Impression Ads.txt Dashboard 2019. Ads.txt Industry Dashboard. firstimpression.io. https://adstxt.firstimpression.io/.
[38] Phillipa Gill, Vijay Erramilli, Augustin Chaintreau, Balachander Krishnamurthy, Konstantina Papagiannaki, and Pablo Rodriguez. 2013. Follow the Money: Understanding Economics of Online Aggregation and Advertising. In *Proc. of IMC*.
[39] Google Ads.txt Manager [n. d.]. Declare authorized sellers with ads.txt. Google. https://support.google.com/admanager/answer/7441288?hl=en.
[40] Google Enforcement 2018. Google Strengthens Ads.txt Enforcement. Ad Exchanger. https://adexchanger.com/ad-exchange-news/google-strengthens-ads-txt-enforcement/.
[41] OpenRTB Working Group. 2019. IAB Tech Lab ads.txt Specification Version 1.0.2. IAB Tech Lab. https://iabtechlab.com/wp-content/uploads/2019/03/IAB-OpenRTB-Ads.txt-Public-Spec-1.0.2.pdf.
[42] OpenRTB Working Group. 2019. IAB Tech Lab Authorized Sellers for Apps (app-ads.txt) Version 1.0. IAB Tech Lab. https://iabtechlab.com/wp-content/uploads/2019/03/app-ads.txt-v1.0-final-.pdf.
[43] OpenRTB Working Group. 2019. IAB Tech Lab Sellers.json DRAFT FOR PUBLIC COMMENT v1.0. IAB Tech Lab. https://iabtechlab.com/wp-content/uploads/2019/04/Sellers.json-Public-Comment-April-11-2019.pdf.
[44] Saikat Guha, Bin Cheng, and Paul Francis. 2010. Challenges in Measuring Online Advertising Systems. In *Proc. of IMC*.
[45] Hamed Haddadi. 2010. Fighting Online Click-fraud Using Bluff Ads. *SIGCOMM Comput. Commun. Rev.* 40, 2 (April 2010), 21–25.
[46] James Hercher. 2018. Google Strengthens Ads.txt Enforcement. ad exchanger. https://adexchanger.com/ad-exchange-news/google-strengthens-ads-txt-enforcement/.
[47] Hyphbot White Paper 2017. How Adform Discovered HyphBot. AdForm. https://site.adform.com/media/85132/hyphbot_whitepaper_.pdf.
[48] IAB. [n. d.]. A reference implementation in python of a simple crawler for Ads.txt. Github. https://github.com/InteractiveAdvertisingBureau/adstxtcrawler.
[49] IAB Ad Fraud Report 2015. What is an untrustworthy supply chain costing the US digital advertising industry? Interactive Advertising Bureau (IAB). https://www.iab.com/wp-content/uploads/2015/11/IAB_EY_Report.pdf.
[50] Integral Ads Domain Spoofing 2015. The four types of domain spoofing. Integral Ads. https://insider.integralads.com/the-four-types-of-domain-spoofing/.
[51] Vishveshwar Jatain. 2019. What is Domain Spoofing? Ad PushUp. https://www.adpushup.com/blog/what-is-domain-spoofing/.

[52] Martijn Koster. 2007. A Standard for Robot Exclusion. http://www.robotstxt.org/orig.html.

[53] Balachander Krishnamurthy, Delfina Malandrino, and Craig E. Wills. 2007. Measuring Privacy Loss and the Impact of Privacy Protection in Web Browsing. In *Proc. of the Workshop on Usable Security*.

[54] Balachander Krishnamurthy, Konstantin Naryshkin, and Craig Wills. 2009. Privacy Diffusion on the Web: A Longitudinal Perspective. In *Proc. of WWW*.

[55] Balachander Krishnamurthy and Craig Wills. 2011. Privacy leakage vs. Protection measures: the growing disconnect. In *Proc. of W2SP*.

[56] IAB Tech Lab. 2018. IAB TECH LAB LAUNCHES PHASE TWO OF OPENRTB 3.0 PUBLIC COMMENT, RELEASING TECH SPECIFICATIONS & KICKING-OFF BETA TESTS. IAB. https://iabtechlab.com/press-releases/openrtb-3-0-beta/.

[57] Tobias Lauinger, Abdelberi Chaabane, Sajjad Arshad, William Robertson, Christo Wilson, and Engin Kirda. 2017. Thou Shalt Not Depend on Me: Analysing the Use of Outdated JavaScript Libraries on the Web. In *Proc of NDSS*.

[58] Pedro Giovanni Leon, Blase Ur, Yang Wang, Manya Sleeper, Rebecca Balebako, Richard Shay, Lujo Bauer, Mihai Christodorescu, and Lorrie Faith Cranor. 2013. What Matters to Users?: Factors That Affect Users' Willingness to Share Information with Online Advertisers. In *Proc. of the Workshop on Usable Security*.

[59] Adam Lerner, Anna Kornfeld Simpson, Tadayoshi Kohno, and Franziska Roesner. 2016. Internet Jones and the Raiders of the Lost Trackers: An Archaeological Study of Web Tracking from 1996 to 2016. In *Proc. of USENIX Security Symposium*.

[60] LUMA Partners LLC 2019. Display LUMAscape. LUMA Partners LLC. https://lumapartners.com/content/lumascapes/display-ad-tech-lumascape/.

[61] Miguel Malheiros, Charlene Jennett, Snehalee Patel, Sacha Brostoff, and Martina Angela Sasse. 2012. Too Close for Comfort: A Study of the Effectiveness and Acceptability of Rich-media Personalized Advertising.

[62] Jonathan R. Mayer and John C. Mitchell. 2012. Third-Party Web Tracking: Policy and Technology. In *Proc. of IEEE Symposium on Security and Privacy*.

[63] Aleecia M. McDonald and Lorrie Faith Cranor. 2010. Americans' Attitudes About Internet Behavioral Advertising Practices. In *Proc. of WPES*.

[64] Methbot Operation 2016. WhiteOps - The Methbot Operation. WhiteOps. https://www.whiteops.com/methbot.

[65] Ahmed Metwally, Divyakant Agrawal, and Amr El Abbadi. 2005. Duplicate detection in click streams. In *Proc. of WWW*.

[66] Ahmed Metwally, Divyakant Agrawal, and Amr El Abbadi. 2007. Detectives: detecting coalition hit inflation attacks in advertising networks streams. In *Proc. of WWW*.

[67] Brad Miller, Paul Pearce, Chris Grier, Christian Kreibich, and Vern Paxson. 2011. What's Clicking What? Techniques and Innovations of Today's Clickbots.

[68] Hamid Nazerzadeh, Amin Saberi, and Rakesh Vohra. 2008. Dynamic Cost-per-action Mechanisms and Applications to Online Advertising. In *Proc. of WWW*.

[69] Lukasz Olejnik. 2018. Enhancing user transparency in online ads ecosystem with site self-disclosures. https://lukaszolejnik.com/adstxt-transparency.pdf.

[70] Lukasz Olejnik, Tran Minh-Dung, and Claude Castelluccia. 2014. Selling off Privacy at Auction. In *Proc of NDSS*.

[71] OpenX Enforcement 2018. OpenX Announces New Ads.txt Policy Banning All Unauthorized Resellers. Business Wire. https://www.businesswire.com/news/home/20180131005710/en/OpenX-Report-Finds-Ads.txt-Adoption-Accelerating-Majority.

[72] Panagiotis Papadopoulos, Nicolas Kourtellis, Pablo Rodriguez, and Nikolaos Laoutaris. 2017. If you are not paying for it, you are the product: How much do advertisers pay for your personal data?. In *Proc. of IMC*.

[73] Paul Pearce, Vacha Dave, Chris Grier, Kirill Levchenko, Saikat Guha, Damon McCoy, Vern Paxson, Stefan Savage, and Geoffrey M. Voelker. 2014. Characterizing Large-Scale Click Fraud in ZeroAccess. In *Proc. of CCS*.

[74] Tim Peterson. 2018. Ads.txt has gained adoption, but 19 percent of advertisers still havenâĂŹt heard of it. Digiday. https://digiday.com/media/state-ads-txt-5-charts/.

[75] Pixalate Ads.txt Adoption 2018. Ads.txt adoption: IABâĂŹs program grows 5.4x in 2018. Pixalate. https://blog.pixalate.com/ads-txt-adoption-trends.

[76] Pixalate Ads.txt Fraud Reduction 2018. Ads.txt reduces ad fraud by 10fraud rates persist. Pixalate. https://blog.pixalate.com/does-ads-txt-reduce-ad-fraud.

[77] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej KorczyÅĎski, and Wouter Joosen. 2019. Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. In *Proc of NDSS*.

[78] PwC Online Advertising Forecast 2018. US Online and Traditional Media Advertising Outlook, 2018-2022. Marketing Charts. https://www.marketingcharts.com/featured-104785.

[79] Abbas Razaghpanah, Rishab Nithyanand, Narseo Vallina-Rodriguez, Srikanth Sundaresan, Mark Allman, Christian Kreibich, and Phillipa Gill. 2018. Apps, Trackers, Privacy and Regulators: A Global Study of the Mobile Tracking Ecosystem. In *Proc of NDSS*.

[80] Neal Richter. 2017. Helping the industry prevent the sale of counterfeit inventory with ads.txt. IAB Tech Lab. https://iabtechlab.com/blog/helping-industry-prevent-sale-of-counterfeit-inventory-with-ads-txt/.

[81] rubiconProject Enforcement 2018. BUYERS MUST STAND UP FOR ADS.TXT. rubiconProject. https://rubiconproject.com/insights/technology/buyers-must-stand-up-for-ads-txt/.

[82] Walter Rweyemamu, Tobias Lauinger, Christo Wilson, William Robertson, and Engin Kirda. 2019. Clustering and the Weekend Effect: Recommendations for the Use of Top Domain Lists in Security Research.

[83] Quirin Scheitle, Oliver Hohlfeld, Julien Gamba, Jonas Jelten, Torsten Zimmermann, Stephen D. Strowes, and Narseo Vallina-Rodriguez. 2018. A Long Way to the Top: Significance, Structure, and Stability of Internet Top Lists. In *Proc. of IMC*.

[84] Samuel Scott. 2016. The $8.2 Billion Adtech Fraud Problem That Everyone Is Ignoring. TechCrunch. https://techcrunch.com/2016/01/06/the-8-2-billion-adtech-fraud-problem-that-everyone-is-ignoring/.

[85] Jarrad Shearer. 2011. Trojan.Zeroaccess. Symantec). https://www.symantec.com/security-center/writeup/2011-071314-0410-99.

[86] George P. Slefo. 2016. Ad Fraud Will Cost $7.2 Billion in 2016, ANA Says, Up Nearly $1 Billion. AdAge. https://adage.com/article/digital/ana-report-7-2-billion-lost-ad-fraud-2015/302201.

[87] Kevin Springborn and Paul Barford. 2013. Impression Fraud in On-line Advertising via Pay-Per-View Networks. In *Proc. of USENIX Security Symposium*.

[88] Brett Stone-Gross, Ryan Stevens, Apostolis Zarras, Richard Kemmerer, Chris Kruegel, and Giovanni Vigna. 2011. Understanding Fraudulent Activities in Online Ad Exchanges. In *Proc. of IMC*.

[89] Kurt Thomas, Elie Bursztein, Chris Grier, Grant Ho, Nav Jagpal, Alexandros Kapravelos, Damon Mccoy, Antonio Nappa, Vern Paxson, Paul Pearce, Niels Provos, and Moheeb Abu Rajab. 2015. Ad Injection at Scale: Assessing Deceptive Advertisement Modifications. In *Proc. of IEEE Symposium on Security and Privacy*.

[90] Sam Tingleff. 2019. The Three Deadly Sins of ads.txt and How Publishers Can Avoid Them . IAB Tech Lab. https://iabtechlab.com/blog/the-three-deadly-sins-of-ads-txt-and-how-publishers-can-avoid-them/.

[91] Michael Carl Tschantz, Serge Egelman, Jaeyoung Choi, Nicholas Weaver, and Gerald Friedland. 2018. The Accuracy of the Demographic Inferences Shown on Google's Ad Settings. In *Proc. of WPES*.

[92] Joseph Turow, Michael Hennessy, and Nora Draper. 2015. The Tradeoff Fallacy: How Marketers Are Misrepresenting American Consumers And Opening Them Up to Exploitation. Report from the Annenberg School for Communication. https://www.asc.upenn.edu/sites/default/files/TradeoffFallacy_1.pdf.

[93] Narseo Vallina-Rodriguez, Jay Shah, Alessandro Finamore, Yan Grunenberger, Konstantina Papagiannaki, Hamed Haddadi, and Jon Crowcroft. 2012. Breaking for Commercials: Characterizing Mobile Advertising. In *Proc. of IMC*.

[94] Giridhari Venkatadri, Yabing Liu, Athanasios Andreou, Oana Goga, Patrick Loiseau, Alan Mislove, and Krishna P. Gummadi. 2018. Privacy Risks with Facebook's PII-based Targeting: Auditing a Data Broker's Advertising Interface. In *Proc. of IEEE Symposium on Security and Privacy*.

[95] WhoTracksMe Data [n. d.]. WhoTracks.me - Bringing Transparency to Online Tracking. Cliqz GmbH. https://whotracks.me/.

[96] Craig E. Wills and Can Tatar. 2012. Understanding What They Do with What They Know. In *Proc. of WPES*.

[97] Shuai Yuan, Jun Wang, and Xiaoxue Zhao. 2013. Real-time Bidding for Online Advertising: Measurement and Analysis. In *Proc. of ADKDD*.

[98] Apostolis Zarras, Alexandros Kapravelos, Gianluca Stringhini, Thorsten Holz, Christopher Kruegel, and Giovanni Vigna. 2014. The Dark Alleys of Madison Avenue: Understanding Malicious Advertisements. In *Proc. of IMC*.

[99] Linfeng Zhang and Yong Guan. 2008. Detecting Click Fraud in Pay-Per-Click Streams of Online Advertising Networks. In *Proc. of ICDCS*.